LOCO implementation for FNAL Booster

M. McAteer, A. Petrenko, E. Gianfelice

April 30, 2013

1 Introduction

In this note the current implementation of LOCO (Linear Optics from Closed Orbit) [1] which has been successfully applied to the Fermilab Booster in the recent past [2] is described. The closed orbit response to dipole corrector variations is acquired by the BPMs and values for normal and skew quadrupole error as well as roll angle and calibrations of BPMs and dipole correctors are found by fitting the measurements. The procedure for acquiring data and set the corrections has been created by M. McAteer, while the LOCO implementation is due to A. Petrenko.
The machine model is obtained by merging the nominal machine description with the actual parameter values. LOCO computes the values of the unknown quadrupole errors which added to the known model reproduce the machine response to the variation of corrector dipoles and beam energy.
The errors are represented by 1 μm long fictitious lenses located at the position of the 48 skew and 48 normal corrector quadrupoles. Their values may be used for correcting coupling and β-beating. However at the moment the control system does not support separate ramps, that is it treats those corrector elements as there would be just one power supply for the SQS (or QS) and one for the SQL (or QL) magnets. This adds some complications to the procedure and has as consequence that possible corrections cannot be easily integrated in the usual Booster operation.
The whole data acquisition and machine handling is done through acl scripts under **LOCO** in the **sequencer** (B116) (see Fig. 1).

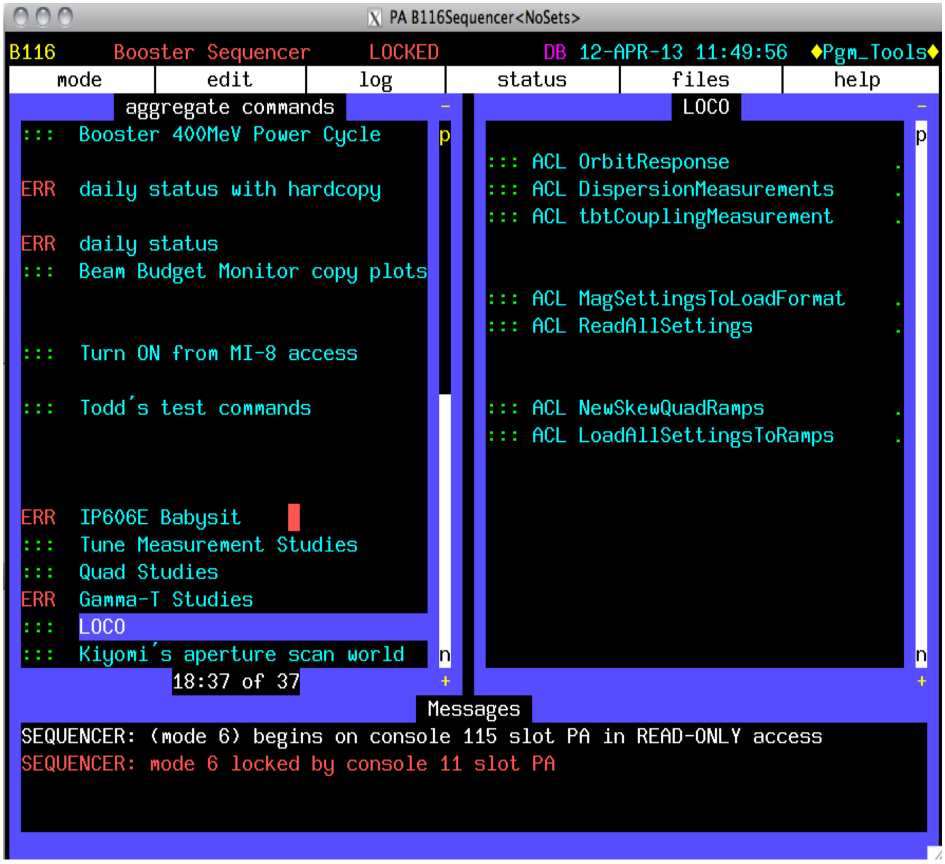

Figure 1: Acl scripts available under LOCO.

2 Settings saving and restoring

A bunch of acl scripts are used for saving/restoring settings in the (unsupported) **"multiple ramps"** modus. They write into user named files and read from fixed named files.
MagSettingsToLoadFormat saves the machine status into the directory **/usr/local/cbs_files/cns_write/booster/LOCO/machineSettings/SettingsToLoad/**.
LoadAllSettingsToRamps reads in the same directory the files

QLRamps.txt	QLOffsets.txt	QSRamps.txt	QSOffsets.txt
SQLRamps.txt	SQLOffsets.txt	SQSRamps.txt	SQSOffsets.txt
SXLRamps.txt	SXLOffsets.txt	SXSRamps.txt	SXSOffsets.txt
SSLRamps.txt	SSLOffsets.txt	SSSRamps.txt	SSSOffsets.txt
HLRamps.txt	HLOffsets.txt	HSRamps.txt	HSOffsets.txt
VLRamps.txt	VLOffsets.txt	VSRamps.txt	VSOffsets.txt

previously created by MagSettingsToLoadFormat and load them to **table 3**. The script **ReadAllSettings** collect the actual settings for completing the LOCO model and writes them to file into the directory **/usr/local/cbs_files/cns_write/booster/LOCO/machineSettings/**.

3 Data acquisition

Data acquisition is done by launching the acl script **OrbitResponse** which steers the corrector dipoles and **DispersionMeasurements** which steers the RF phase. The files **ORMConfig.txt** and **DispConfig.txt** under **/usr/local/cbs_files/cns_write/booster/LOCO/** contain the parameters values. The meaning of the parameters in the ORMConfig.txt and DispConfig.txt rows and their customary values are shown in Table 1 and Table 2 respectively.
The parameters for reading the orbit must be set through **B40** (see Fig. 2). Meaning and values are shown in Table 3.

plane of measurement	0 for horiz. scan, 1 for vertical
first corrector timepoint slot to change	2
number of corrector time slots to change	5
number of orbits (ie time points during ramp) to read	32
number of times to repeat each measurement	6
first machine period to measure	1
number of periods to measure	24
size of I vs. t bump (A)	2 for horiz. scan, 3 for vertical

Table 1: ORMConfig.txt parameters.

first ROF slot to change before transition	10
number of ROF slots to change before transition	22
first ROF slot to change after transition	40
number of ROF slots to change after transition	14
number of orbits to read	32
number of times to repeat each measurement	6
list of ROF values (w/o commas)	3 2 1 0 -1

Table 2: DispConfig.txt parameters

TURNS/ORBIT	=	10
NUM OF ORBITS[a]	=	32
TURNS TO SKIP	=	490
BEGINNING TURN	=	500

[a] not smaller than the value set in ORMConfig.txt and DispConfig.txt

Table 3: B40 parameters

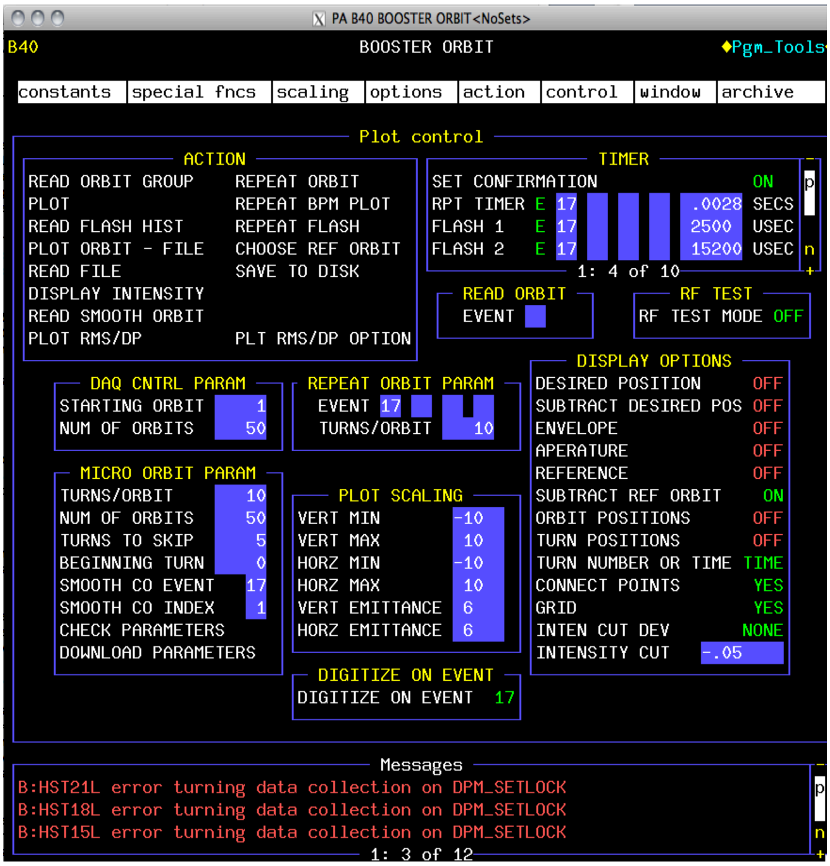
PA B40 BOOSTER ORBIT<NoSets>
B40
BOOSTER ORBIT
Pgm_Tools
constants
special fncs
scaling
options
action
control
window
archive
Plot control
ACTION
READ ORBIT GROUP
PLOT
READ FLASH HIST
PLOT ORBIT - FILE
READ FILE
DISPLAY INTENSITY
READ SMOOTH ORBIT
PLOT RMS/DP
REPEAT ORBIT
REPEAT BPM PLOT
REPEAT FLASH
CHOOSE REF ORBIT
SAVE TO DISK
PLT RMS/DP OPTION
TIMER
SET CONFIRMATION ON
RPT TIMER E 17 .0028 SECS
FLASH 1 E 17 2500 USEC
FLASH 2 E 17 15200 USEC
1: 4 of 10
READ ORBIT
EVENT
RF TEST
RF TEST MODE OFF
DAQ CNTRL PARAM
STARTING ORBIT 1
NUM OF ORBITS 50
REPEAT ORBIT PARAM
EVENT 17
TURNS/ORBIT 10
DISPLAY OPTIONS
DESIRED POSITION OFF
SUBTRACT DESIRED POS OFF
ENVELOPE OFF
APERATURE OFF
REFERENCE OFF
SUBTRACT REF ORBIT ON
ORBIT POSITIONS OFF
TURN POSITIONS OFF
TURN NUMBER OR TIME TIME
CONNECT POINTS YES
GRID YES
INTEN CUT DEV NONE
INTENSITY CUT -.05
MICRO ORBIT PARAM
TURNS/ORBIT 10
NUM OF ORBITS 50
TURNS TO SKIP 5
BEGINNING TURN 0
SMOOTH CO EVENT 17
SMOOTH CO INDEX 1
CHECK PARAMETERS
DOWNLOAD PARAMETERS
PLOT SCALING
VERT MIN -10
VERT MAX 10
HORZ MIN -10
HORZ MAX 10
VERT EMITTANCE 6
HORZ EMITTANCE 6
DIGITIZE ON EVENT
DIGITIZE ON EVENT 17
Messages
B:HST21L error turning data collection on DPM_SETLOCK
B:HST18L error turning data collection on DPM_SETLOCK
B:HST15L error turning data collection on DPM_SETLOCK
1: 3 of 12

Figure 2: B40.

4 Fit with LOCO

4.1 Data pre-processing

The acquired orbit data are written into ascii files in the directory **/usr/local/cbs_files/cns_write/booster/LOCO/data/**. They must be processed for producing the values of dz_i/dI_k and $dz_i/d\epsilon_j$, ϵ being the relative energy error. LOCO has been implemented on heimdall. For reducing the number of machines and data transfers the preprocessing is done with (fortran) programs which run on clx under /export/home1/eliana/booster_orm/fit_orm.exe and /export/home1/eliana/booster_disp/fit_disp.exe. They produce the input files D.sdds and ORM.sdds for LOCO in ascii format.

4.2 Fit

The theoretical machine response is computed using the ANL optics code Elegant [3] which has been installed on heimdall together with the SDDSToolkit [3]. For sake of clarity the implementation by A. Petrenko is described.
The tcl scripts used for the fit are contained in the directory ORM_fit. This directory contains the subdirectories:

- the ORM_fit/data which contains the measured data and the script element_settings2sdds.tcl for reading the actual machine settings;
- ORM_fit/results contains the results;
- ORM_fit/ele contains the files ORM.ele and twiss.ele needed by elegant. The ideal machine is contained in the file ORM_fit/machine.lte.

As said in Section 3, the machine model is integrated with the actual settings. The file ORM_fit/data/MagnetSettings.txt produced by ReadAllSettings is read by ORM_fit/data/element_settings2sdds.tcl and converted into a sdds binary file. The measurement files are ORM_fit/data/ORM.sdds and ORM_fit/D.sdds. For speeding up computation those files are converted into binary files once transferred on heimdall, by using sddsconvert from the SDDSToolkit. The conversion is obtained by issuing

sddsconvert *ascii_filename binary_filename* -binary

The whole ramp fit performed by the ORM_fit/fit_wole_ramp.tcl script. Starting values for the fitting parameters are read from ORM_fit/machine_parameters_vs_t.sdds which contains either the nominal values or those found by a previous LOCO run. Those values are used for initializing the parameters in the first time slot. The newly computed values for slot n are used for the subsequent slot $n + 1$. The results are written into the binary file ORM_fit/results/machine_parameters_vs_t.sdds which may be converted to a ascii file by

sddsconvert *binary_filename* *ascii_filename* -ascii

The file contains

- dipole correctors calibration
- dipole correctors roll angle (rad)
- BPMs gain
- BPMs roll angle (rad)
- strength of the normal quadrupole errors in m^{-2}
- strength of the skew quadrupole errors in m^{-2}

5 Using fit results for optics correction

5.1 How to use the results

The values found by LOCO for the errors may be used directly for correcting the optics. The strength inferred by the measured response, $K^{(0)}_{meas}$, is given by the unknown errors plus the known model

$$K^{(0)}_{meas} = K_{err} + K^{(0)}_{model} = K^{(0)}_{fit} + K^{(0)}_{model}$$

If the fit is well representing the unknown errors it is $K_{err} = K^{(0)}_{fit}$. If the model is changed by

$$\Delta K^{(0)}_{model} = K_{design} - (K^{(0)}_{fit} + K^{(0)}_{model})$$

(K_{design} being the design optics), the new model $K^{(1)}_{model}$ is given by

$$K^{(1)}_{model} = K^{(0)}_{model} + \Delta K^{(0)}_{model} = K^{(0)}_{model} + [K_{design} - (K^{(0)}_{fit} + K^{(0)}_{model})] = K_{design} - K^{(0)}_{fit}$$

and therefore the new measured response will be

$$K^{(1)}_{meas} = K_{err} + K^{(1)}_{model} = K_{err} + K_{design} - K^{(0)}_{fit} = K_{design}$$

if the fit is well representing the unknown errors.
In particular for the coupling it is

$$K^{(1)}_{model} = -K^{(0)}_{fit}$$

as the Booster is by design uncoupled.

5.2 Processing the fit results

The data must be translated into ramps for the 48 normal and 48 skew quadrupole corrector circuits. At the moment this is done by a Mathematica [4] notebook by M. McAteer. This notebook reads the skew quadrupole errors from machine_parameters.sdds (ascii format), converts them to magnet current values, interpolates the values at the existing skew magnet ramp time points and writes new ramps. The time breakpoints are read from a second file. The two files produced for the SQS and SQL corrector must be copied into
/usr/local/cbs_files/cns_write/booster/LOCO /NewSQSRamps.txt and
/usr/local/cbs_files/cns_write/booster/LOCO/NewSQLRamps.txt in order to be read by the NewSkewQuadRamps script launched by the sequencer.

References

[1] J. Safranek, NIM A388, 27 (1997).

[2] http://accelconf.web.cern.ch/accelconf/IPAC2012/papers/tuppc040.pdf

[3] http://www.aps.anl.gov/Accelerator_Systems_Division/Accelerator_Operations_Physics/software.shtml

[4] Wolfram Research, Inc., Mathematica, Version 8.0, Champaign, IL (2010).